

БИБЛИОТЕКА МЕТОДОВ НЕЛИНЕЙНОЙ ДИНАМИКИ В ПАКЕТЕ MATLAB

В книге «Хаос и порядок на рынках капитала» Э. Петерс предложил использовать фракталы для анализа динамики стоимости ценных бумаг. Автор выбрал ряд известных методов анализа нелинейных систем и использовал для временных рядов, описывающих изменение во времени некоторой величины системы. Он привел описания счетных алгоритмов этих методов. Алгоритмы широко применяются в разных областях, например, анализ динамики количества солнечных пятен, стоков рек, цен акций и так далее. Но хорошей реализации комплексной библиотеки в Интернете найти не удалось. В докладе описывается библиотека, реализующая следующие алгоритмы:

- восстановление фазового пространства – алгоритм Такенса,
- расчет корреляционной размерности аттрактора,
- R/S анализ (метод нормированного размаха),
- расчет показателя Херста, основанный на R/S анализе,
- имитация случайного блуждания,
- вычисление наибольшего показателя Ляпунова.

Реализация описанных алгоритмов допустима и на языках высокого уровня, в электронных таблицах и так далее; здесь для реализации библиотеки используется язык математического пакета MATLAB. Этот пакет позволяет широко использовать стандартные математические средства и сфокусировать усилия на разработке алгоритмов, а не реализации базовых элементов, например, линейной аппроксимации.

Все алгоритмы реализованы в виде отдельных функций, и их можно использовать в более сложных программах, как и другие средства MATLAB. В общем случае основными входными данными для всех алгоритмов является временной ряд данных – одномерный массив.

Состав библиотеки

Функция: `CorrelationDimension(data, m, J, minDist, numStep)`

Назначение: Вычисление корреляционной размерности

Аргументы: `data` – временной ряд

`m` – размерность восстанавливаемого пространства

`J` – временная задержка

`minDist` – начальное расстояние

`numStep` – количество шагов

Результат: Число, значение корреляционной размерности

Функция: `FractionalBrownianMotion(data, H, M, n)`

Назначение: Имитация смещенного случайного блуждания

Аргументы: `data` – ряд случайных чисел

Н – показатель Хёрста

М – длина памяти

n – интервал памяти

Результат: Массив значений временного ряда случайного блуждания

Функция: `RescaledRange(data, startIndex, countIndex)`

Назначение: Метод нормированного размаха (R/S – анализ)

Аргументы: `data` – временной ряд

`startIndex` – начальное значение

`countIndex` – количество отсчетов

Результат: Число - R/S значение части временного ряда

Функция: `Hurst(data)`

Назначение: Рассчет показателя Херста

Аргументы: `data` – временной ряд

Результат: Число - R/S значение временного ряда

Функция: `LargestLyapunovExp(data, m, J, mperiod, dt, max, min)`

Назначение: Вычисление наибольшего показателя Ляпунова

Аргументы: `data` – временной ряд

`m` – размерность восстанавливаемого пространства

`J` – временная задержка

`mperiod` – наименьший период

`dt` – время эволюции

`max, min` – максимальное и минимальное расстояния

Результат: Число – наибольший показатель Ляпунова

Примеры работы библиотеки

Посмотрим на работу библиотеки. Сгенерируем случайное блуждание, для этого нам нужен ряд гауссовых случайных чисел.

```
> load_gauss.dat;
```

Данные на рис. 1,а. К сожалению, ряд, взятый для экспериментов, не такой большой, поэтому значения длины и интервалов памяти принимаем 10 и с показателем Хёрста 0.7.

```
> fbm = FractionalBrownianMotion(gauss, 0.7, 10, 10, 700);
```

В результате расчетов получаем следующую картинку (рис. 1, б).

```
> h = Hurst(fbm)
```

h =

0.6838

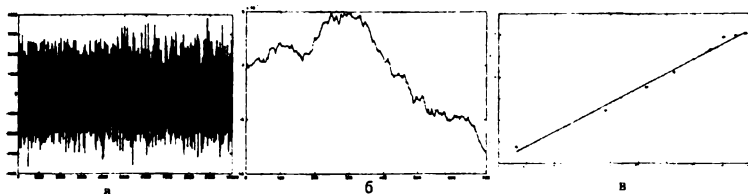


Рис 1. а) случайные числа; б) полученное блуждание; в) расчет показателя Хёрста

Как видно, результаты расчетов близки к тем параметрам, которые были заданы при генерации.

В качестве примеров можно посмотреть рис. 2. На нем приведены результаты имитации случайного блуждания на том же исходном ряде, но с разными показателями Хёрста.

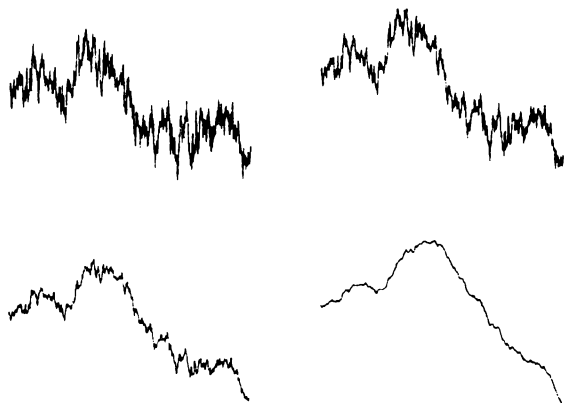


Рис. 2. Имитация случайного блуждания с разными показателями Хёрста (справа налево, сверху вниз 0.1, 0.3, 0.5, 0.9)